Symposium in Honour of Lauri Hella's 60th birthday

Tampere, Finland, 4-6 July 2018

On Fragments of Higher Order Logics that on Finite Structures Collapse to a Lower Order

José María Turull-Torres

Universidad Nacional de La Matanza, Argentina

In-progress joint work with **Flavio Ferrarotti** and **Senén González**.

Contents:

Two Motivating Examples in Third Order Logic

Example 1 in HO^3: Hypercube graphs

An *n-hypercube graph* $\mathbf{Q}_n$, is an undirected graph whose vertices are binary n-tuples. Two vertices of $\mathbf{Q}_n$ are adjacent iff they differ in exactly one bit.

Note that we can build an $(n+1)$-cube $\mathbf{Q}_{n+1}$ starting with two isomorphic copies of an n-cube $\mathbf{Q}_n$ and adding edges between corresponding vertices.

That is, *multiplying* an n-cube graph by K_2.

Using this fact, we can define in TO (HO^3) the class of *hypercube graphs*, by saying that:

- there is a *sequence* of graphs (i.e., a *third order linear digraph*, where every TO node is an undirected (SO) graph)

- which *starts* with the graph K_2, *ends* with a graph which is *equal* to the input graph, and such that

- every graph G_2 in the sequence results from finding two *total, injective* functions f_1, f_2 from the previous graph G_1, so that

– f_1 and f_2 induce in G_2 two *isomorphic copies* of G_1,

– the *images* of those functions define a *partition* in the vertex set of G_2, and

– there is an *edge* in G_2 between the *images* $f_1(x)$ and $f_2(x)$ of *every* node x in G_1.

Actually, the expressive power of HO^3 *is not required* to characterize hypercube graphs, since they can be recognized in NP, and hence in ESO.

Nevertheless, to define the class of hypercube graphs in ESO seems to be *more challenging* than to define it in HO^3.

(see the SO formula for the first strategy considered for hypercube graphs in [Ferrarotti, Ren, Turull-Torres, 2014], and Remark 4.1 there, indicating the way to translate it to an ESO formula).

A Second Definition of Hypercube graphs

Another definition of *hypercube graphs* that yields a simple (TO) formula is the following.

We say that there is a proper non empty *subset* V' of the vertex set V of the input graph G, and a (TO) bijective function $\mathcal{F} : V \to \mathcal{P}(V')$ (i.e., the power set of V'), s. t.

for every pair of nodes x and y in G, there is an edge between them

iff

$\mathcal{F}(x)$ can be obtained from $\mathcal{F}(y)$ by *adding* or *removing* a *single* element (note that V' is necessarily of size $\log_2 |V|$).

Note that the corresponding SO formula is *not so intuitive* (see [Ferrarotti, Ren, Turull-Torres, 2014]).

The *SO* formula that expresses the second strategy is in the class $\sum_2^1$.

The existence of a formula in $\sum_1^1$ that expresses *this* strategy is *unlikely*, since we must express that *every subset* S of V is identified with some node in the graph.

Example 2 in HO^3: Formula-Value Query

Given a propositional formula φ in the constants $\{F, T\}$, represented as a word model, decide whether it is true.

- There is a *sequence* $\mathcal{S}$ of propositional formulas represented as word models.
- $\mathcal{S}$ starts with φ and ends with the formula "T".

- Every formula φ_i in $\mathcal{S}$ (except the first) results from the previous formula φ_{i-1} by either:
  - Application to φ_{i-1} of *one* of $\vee$, $\wedge$ and $\neg$ which is ready to be evaluated.
    - Like in "$(T \wedge F)$".
  - Or elimination of *one* pair of redundant parenthesis in φ_{i-1}.
    - Like in "$((T))$".

- Formula-Value query is in DLOGSPACE [Beaudry, Pierre McKenzie, 1992].
- DLOGSPACE $\subseteq$ P $\subseteq$ NP $=$ $\exists$SO.
- Nevertheless, to define these queries in $\exists$SO seems to be more challenging than in TO (see [Ferrarotti, Ren, Turull-Torres, 2014]).

Note that in the two examples in HO^3 the *size of the valuating relations* for the TO variables that make the formulas true, is *polynomial* (actually *logarithmic* and *linear*, respectively) in the size of the input structure.

On the other hand, if we consider the query SATQBF (see below), we can express it in EHO^3, since the problem is PSPACE complete, and it is known that EHO^3 is powerful enough as to characterize every problem in PSPACE.

Note that the existence of an SO formula that expresses SATQBF is very *unlikely*, since $SO =$ PH, and it is strongly *conjectured* that PH $\subset$ PSPACE.

0: Higher Order Logics

$(\mathrm{HO^i})$

Higher Order Variables
Types

- A first order *variable type* is $\tau^1 = 0$,
- a second order *variable type* is $\tau^2 \geq 1$, i.e., its arity,
- for $i \geq 3$, an i-th order *variable type* is a sequence of types of orders $1 \leq j_1, \ldots, j_s \leq i-1$, $\tau^i = (\tau_1^{j_1}, \ldots, \tau_s^{j_s})$, with $s \geq 1$.

W.l.o.g., we assume that at least one of the types $\tau_1^{j_1}, \ldots, \tau_s^{j_s}$ is of order $i-1$.

In the alphabet of a *Higher Order Logic of order* i, HO^i, for every *order* $2 \leq j \leq i$, and for every *variable type* τ, we add to FO a countably infinite set of *relation variables* $\mathcal{X}_1^{j,\tau}, \mathcal{X}_2^{j,\tau}, \ldots$

We use calligraphic letters like $\mathcal{X}^i$ and $\mathcal{Y}^i$ for variables of order $i \geq 3$, upper case letters like X and Y for second order variables, and lower case letters like x and y for first order variables.

Besides the atomic formulas in FO and SO, in HO^i we can use *atomic formulas* like the following:

If $\mathcal{X}$ is a relation variable of order j, for some $3 \leq j \leq i$, and of relation type τ, for some $\tau = (\rho_1, \ldots, \rho_s)$, with $\rho_1, \ldots, \rho_s$ being types of orders $\leq j-1$, and $\mathcal{Y}_1, \ldots, \mathcal{Y}_s$ are relation variables of orders and types according $\rho_1, \ldots, \rho_s$, respectively, then $\mathcal{X}(\mathcal{Y}_1, \ldots, \mathcal{Y}_s)$ is an atomic formula.

Higher Order Relations

Let $s \geq 1$. An *SO* relation of arity s is a relation in the classical sense, i.e., a set of s-tuples of elements of the domain of a given structure.

For an arbitrary $i \geq 3$, a *relation of order j* of relation type $\tau = (\rho_1, \ldots, \rho_s)$, is a set of *$s$-tuples of relations* of orders and types according $\rho_1, \ldots, \rho_s$, respectively.

W.l.o.g., and for the sake of simplicity, we assume that the width of a higher order relation is *propagated downwards*, i.e., the relations of order $i-1$ which form the s-tuples for a relation of order i, are themselves of width s, and so on, all the way down to the SO relations, which are also of arity s.

We define $exp(0) = O(n^{O(1)})$, and for $i \geq 1$

$$exp(i) = 2^{exp(i-1)}$$

That is, $exp(i)$ is a *hyper exponential function*, which we define as a stack of i exponents 2, and then $O(n^{O(1)})$ as the topmost exponent.

(*) actually the i exponents should be $O(1)$, but we write 2 for simplicity.

Maximum Cardinalities of HO Relations

- SO relations: $\leq n^{O(1)}$;
- TO relations: $\leq 2^{O(n^{O(1)})}$;
- HO^4 relations: $\leq 2^{(2^{O(n^{O(1)})})} = exp(2)$;
- HO^5 relations: $\leq 2^{(2^{(2^{O(n^{O(1)})})})} = exp(3)$;
- ...
- HO^{i} relations: $\leq exp(i-2)$.

Σ^i_j

Let $i, j \geq 1$, as it is usual in classical Logic we denote by Σ^i_j the class of formulas $\varphi \in HO^{i+1}$ of the form

$$\exists \mathcal{X}_{11} \ldots \exists \mathcal{X}_{1s_1} \forall \mathcal{X}_{21} \ldots \forall \mathcal{X}_{2s_2} \exists \mathcal{X}_{31}$$

$$\ldots \exists \mathcal{X}_{3s_3} \ldots Q \mathcal{X}_{j1} \ldots Q \mathcal{X}_{js_j} (\psi)$$

where $\psi \in HO^i$, Q is either $\exists$ or $\forall$, depending on whether j is odd or even, respectively.

That is, Σ^i_j is the class of HO^{i+1} formulas with $j-1$ alternations of quantifiers blocks of variables of order $i+1$, starting with an existential quantifier.

Analogously, we define the classes of formulas Π^i_j.

Expressibility of Higher Order Logics

[Hella, Turull-Torres, 2006]

1. For every $i \geq 0$, let

$$\mathrm{NEXPH}_i^0 =$$

$$\mathrm{NTIME}(exp(i))$$

2. For every $j \geq 1$, let

$$\mathrm{NEXPH}_i^j = \mathrm{NEXPH}_i^{0^{\Sigma_{j-1}^p}}$$

Recall that $\Sigma_1^p = \Sigma_1^1 = \mathrm{NP}$ and $\Sigma_0^p = \mathrm{P}$.

[Hella, Turull-Torres, 2006]

- for $i, j \geq 1$: $\Sigma^i_j = \mathrm{NEXPH}^{j-1}_{i-1}$.

  That is, a stack of $i-1$ exponents 2, and then $O(n^{O(1)})$ as the topmost exponent, plus an oracle in Σ^p_{j-1}.

- for $i, j \geq 1$: $\Pi^i_j = co\mathrm{-NEXPH}^{j-1}_{i-1}$.

Fragments of HO^{i} with *Small* Valuating Relations

We have seen above sketches of HO^3 formulas for the queries *Hypercube graphs* and *Formula-Value.*

As we pointed out then, the expressive power of HO^3 *is not actually required* for any of them.

Could we...?

Could we take advantage of the *much higher* expressibility and simplicity of HO^3,

•and, still

be able to express a query in a *more simple* and *intuitive* way, <u>*though still formal*</u> (*),

• but

without having to pay the price of a *higher complexity* to evaluate the corresponding formulas?

(*) so we can still make use of semi-automatic *theorem proving* (see below).

Note that by the results given above

$$ESO = \text{NTIME}(n^{O(1)}) \subseteq \text{DTIME}(2^{n^{O(1)}}),$$

while

$$ETO = \text{NTIME}(2^{n^{O(1)}}) \subseteq \text{DTIME}(2^{2^{n^{O(1)}}}).$$

What is good about HO^i?

For all $i \geq 2$, HO^{i+1} provides two important features:

- *exponentially bigger* auxiliary relations than HO^i;
- *nesting* of relations, like in $(i+1)$*-th order graphs*, where each node is actually an i-th order graph,

  or

  $(i{+}1)$*-th order PERT networks*, for large and complex projects, where a node may represent a PERT network itself, and the operation of *zooming* in or out allows navigation in depth.

But...

The complexity of the evaluation of an HO^{i+1} query is *exponentially higher* than that of an HO^{i} query (see above).

For instance, for *Existential Fourth Order Logic* queries (Σ^3_1) the complexity is

$$= \bigcup_{c \in N} \mathrm{NTIME}(2^{2^{(n^c)}})$$

While for *Existential Third Order Logic* queries (Σ^2_1) is

$$= \bigcup_{c \in N} \mathrm{NTIME}(2^{(n^c)})$$

What if...?

What happens if we bound the *size* of the i-th order relations to be *polynomial* in the size of the input dbi?

We could still have *nesting*...

Besides being a *requirement* in some applications (like *deep structures* where zoom operations are necessary),

in many cases

- *nesting* provides a *more powerful* language which allows *simpler* and *more intuitive* expressions for a query.

This also happens when using programming languages with *rich data structures* (like OOPL):

- it makes programs much *simpler* and *less error-prone* than using the old Assembler languages of the sixties and seventies.

• This is convenient not only for applications to Databases in the Industry, but also for *Theoretical* research.

• To prove that a query is in the *polynomial hierarchy* (PH), in many cases using higher order constructions in $\mathrm{HO}^{i,P}$ can be much simpler than using SO (see below).

• To prove that a query is in the *poly-logarithmic hierarchy* (PLH), in many cases using higher order constructions in $\mathrm{HO}^{i,plog}(\mathrm{HO}^{<i,plog})$ can be much simpler than using SO^{plog} (see below).

- Is nesting still relevant as to *expressive power*?

1: A General Schema of TO Formulas

Let σ be a relational vocabulary, which may include constant symbols. We define $\mathfrak{T}[\sigma]$ as the class of TO formulas of the form:

$$\exists \mathcal{C}^{\bar{s}} \mathcal{O}^{\bar{s}\bar{s}} \Bigg(\text{TotalOrder}(\mathcal{C}, \mathcal{O}) \wedge \forall G \Big[\big(\text{First}(G) \rightarrow \alpha_{\text{First}}(G) \big) \wedge \big(\text{Last}(G) \rightarrow \alpha_{\text{Last}}(G) \big) \Big] \wedge \forall G_{pred} G_{succ} \Big[\text{Pred}(G_{pred}, G_{succ}) \rightarrow \varphi(G_{pred}, G_{succ}) \Big] \Bigg)$$

where

- $\mathcal{C}$ ranges over TO relations of type $\bar{s} = (i_1, \ldots, i_s)$.
- TotalOrder$(\mathcal{C}, \mathcal{O})$, First$(G)$, Last$(G)$ and Pred(G_{pred}, G_{succ}) denote *fixed* SO formulas.
- $\alpha_{\mathrm{First}}(G)$ and $\alpha_{\mathrm{Last}}(G)$ denote *arbitrary* SO formulas.
- $\varphi(G_{pred}, G_{succ})$ denotes an *arbitrary* SO formula.

This is a very usual, *intuitive*, and convenient schema in the expression of natural properties of finite models.

For a start, it can clearly be used to express the *hypercube* and *formula-value* queries as described above.

Additional examples are provided by the different *relationships* between pairs of undirected *graphs* (G, H) that can be defined as orderings of special sorts (see [Downey, Fellows, 1999]).

Using the schema these relationships can be expressed by defining a set of possible *operations* that can be applied repeatedly to H, until a graph which is isomorphic to G is obtained.

In particular, the following relationships fall into this category:

a) $G \leq_{immersion} H$: G is an *immersion* in H;
b) $G \leq_{top} H$: G is *topologically embedded* or *topologically contained* in H;
c) $G \leq_{minor} H$: G is a *minor* of H;
d) $G \leq_{induced-minor} H$: G is an *induced minor* of H;

Interestingly, in all these cases the *length* of the sequence is at most *linear*.

The operations on graphs needed to define those orderings are:

(E) *delete* an edge,

(V) *delete* a vertex,

(C) *contract* an edge,

(T) degree 2 contraction, or *subdivision removal*,

(L) *lift* an edge.

In particular the set of allowable operations for each of those orderings are:

$\{E, V, L\}$ for $\leq_{immersion}$,

$\{E, V, C\}$ for $\leq_{minor}$,

$\{E, V, T\}$ for $\leq_{top}$,

$\{V, C\}$ for $\leq_{induced-minor}$.

[Ferrarotti, González, Turull-Torres, 2017]

We have the following:

Theorem:

Every TO formula Ψ of the *above schema* $\mathfrak{T}$ can be *translated* into an equivalent SO formula Ψ' whenever the following conditions hold.

1. The sub formulas α_{First}, α_{Last} and φ of Ψ are SO formulas.
2. There is a $d \geq 0$ such that for every valuation v with $v(\mathcal{C}) = \mathcal{R}$, if $\mathbf{A}, v \models \exists \mathcal{O}^{\bar{s}\bar{s}} \psi(\mathcal{C}, \mathcal{O})$, then $|\mathcal{R}| \leq |dom(\mathbf{A})|^d$.

Planarity in Graphs

The classical Kuratowski definition of *planarity*, provides yet another example of a property that can be defined using our schema and also results in a *linearly bounded* sequence of structures.

By Wagner's characterization (see [Bollobás, 2002]) a graph is *planar* if and only if it contains *neither* K_5 *nor* $K_{3,3}$ as a *minor*.

Note that the more *intuitive* construction for *planarity* would be to say that there is *no* transformation process of linear size that arrives to a K_5 or $K_{3,3}$, starting from the input graph and applying in each transition *exactly one* of the operations in $\{E, V, C\}$ above.

If we have the *negation* of a formula in the schema $\mathfrak{T}$, we can use the same translation to SO, and then add a negation in front of the SO formula.

Then we have the following:

Corollary:

The *negation* $\neg\Psi$ of a formula Ψ of the *above schema* $\mathfrak{T}$ can also be *translated* into an equivalent SO formula $\neg\Psi'$ whenever the two conditions of the previous theorem hold.

[Ferrarotti, González, Schewe, Turull-Torres, 2018]

By using the normal form for $(SO+TC^2)$ ([Imm,1999]) the following result is straightforward:

Theorem:
The class of TO formulas of the *above schema* $\mathfrak{T}$ is *equivalent* to the logic $(SO + TC^2)$.

And, hence, equal to PSPACE.

Corollary:
The class of TO formulas of the schema $\mathfrak{T}$ is *closed* under negation.

Translation to Non Det Parallel ASM

[Ferrarotti, González, Schewe, Turull-Torres, 2018]

By using the non deterministic, parallel Abstract State Machine model ([Boerger, 2003]), it is not difficult to prove the following:

Theorem:
Every formula Ψ of the *above schema* $\mathfrak{T}$ can be *systematically* translated to an *equivalent* non deterministic, parallel ASM which *doesn't* use higher order formulas.

Note that for the sake of *easily comprehensible high-level specifications* it is advisable to extend rigorous methods to support also *higher-order logic* and to investigate strategies for *refinement* to first-order.

Theorem provers and Non det Parallel ASM

It is well known that for many cases of ASM's, there are *theorem provers* which allow *semi-automatic theorem proving* support for many cases of ASM rules.

In particular, for *non deterministic parallel* ASM's there are very interesting results.

[Schellhorn,Ernst,Pfhler,Bodenmller, Reif , 2018]

- It is possible to compute an FO formula for each rule that *implies clash-freedom* (*) when provable (it is provable for *many* ASMs that are *used in practice*).

  (*) for each state S a rule r yields an update set $\Delta(S)$, i.e. a (finite) set of (finite) sets of updates. There is a *clash* if there are two updates (l, v_1), (l, v_2) in $\Delta(S)$ with $v_1 \neq v_2$.

(i.e., pairs *location* (i.e., n-ary function symbol and an n-tuple of values), and *value*)

- They give *axioms* that describe the transition relation for *clash-free* ASM rules as *SO* formulas that can be used to *verify* pre/post-condition assertions, and to *derive* properties of ASM's, using *automated theorem provers*.
- They provide a *Calculus* for clash-free ASM rules based on symbolic execution for *deduction*, which can be used for interactive *theorem provers*, like their tool **KIV**.

[Ferrarotti, González, Schewe, Turull-Torres, 2018]

By using higher order logics $\mathrm{HO}^{i,P}$ (see below) the following result is straightforward:

Theorem:
For every ASM *extended* with $\mathrm{HO}^{i,P}$ formulas in its rules, we have an *automatic refinement* of the $\mathrm{HO}^{i,P}$ extended ASM to an *SO extended* ASM.

Once we got the SO extended ASM, we can apply to it the *naïve refinement* strategy consisting on non-deterministically guessing the quantified relation variables.

As *naïve refinements* in a standard way are always possible, we believe that semi-automatic proofs *could be conducted* on such, though *not optimal* refinements.

QBF Solvers

Alternatively, the use of QBF solvers is *worth exploring*.

from "QBF Gallery 2014 (Competition)", in the "QBF Solver Evaluation Portal",
www.qbflib.org/index_eval.php

“Many problems from application domains such as model checking, formal verification or synthesis are PSPACE-complete, and hence could be *encoded* in QBF”.

“*Considerable progress* has been made in QBF solving throughout the past years. However, in contrast to SAT, QBF is *not yet* widely applied to *practical problems* in industrial settings”.

Once we got an *SO formula* ϕ (see below):

- for every model $\mathbf{A}$, there is a translation $f_\phi(\mathbf{A})$ to a QBF *formula* (see [Hella, Turull-Torres, 2006a] for a translation),
- we can then use a QBF *solver*.

2: Downward polynomially bounded Relations

$HO^{i,P}$

An *i-th order relation* $\mathcal{R}$ of type τ in a structure $\mathbf{A}$ is *downward polynomially bounded* (*dpb*) by d if $|\mathcal{R}| \leq |dom(\mathbf{A})|^d$,

and

for all $2 \leq j \leq i-1$, all the j-th order relations that form the tuples of $(j+1)$-th order relations, are in turn *dpb* by d.

For $i \geq 3$ we define $\mathrm{HO}^{i,P}$ as the extension of $\mathrm{HO}^{i-1,P}$, where the i-th order quantifiers restrict the cardinality to be *bounded by a polynomial* that depends on the quantifier.

In the alphabet of $\mathrm{HO}^{i,P}$, for every pair of positive integers d, and j, with $i \geq j \geq 3$, we have:

a j-th order quantifier $\exists^{j,P,d}$

and

for every j-th order type τ, we have countably many j-th order variable symbols $\mathcal{X}^{j,d,\tau}$.

A *valuation* in a structure $\mathbf{A}$ assigns to each i-th order relation variable $\mathcal{X}^{j,d,\tau}$ a *dpb* i-th order relation $\mathcal{R}$ of type τ in A, such that $|\mathcal{R}| \leq |dom(\mathbf{A})|^d$.

For any $2 < j \leq i$, the $\mathrm{HO}^{i,P}$ quantifier $\exists^{j,P,d}$ has the following semantics:

$\mathbf{A} \models \exists^{j,P,d} \mathcal{X}^{j,d,\tau} \varphi(\mathcal{X})$

iff

there is a j-th order relation $\mathcal{R}$ of type τ, such that $\mathbf{A} \models \varphi(\mathcal{X})[\mathcal{R}]$ and $\mathcal{R}$ is *dpb* by d in $\mathbf{A}$.

[Ferrarotti, González, Turull-Torres, 2017]

We have the following:

Theorem:
For all $i \geq 3$, $\mathrm{HO}^{\mathrm{i,P}}$ collapses to SO. Moreover, every formula in $\mathrm{HO}^{\mathrm{i,P}}$ can be *algorithmically translated* to an equivalent SO formula.

Strategy:

Basically, the strategy of the translation is to use a *relational database with referential integrity* to encode *each relation variable* of order ≥ 2.

Let $i \geq k \geq j \geq 2$. For each variable of order k, the db that represents it consists of $2(k-1)$ relations.

For each j-th order variable we have one relation with id's for *tuples of relations* of order $(j-1)$, and one relation for id's of *relations* of order $(j-1)$.

Empty Relations

We must also have in mind that the tuples of relations of any order, can have *empty relations* in some of its components.

Then, the (SO) "*relation*" that we use to store the set of *tuple identifiers* for a relation of type width s, is actually a set of 2^s (SO) relations, one for each possible *combination* of empty relations in such a tuple.

Then, for a given query, we can proceed as follows:

1. • Use an $HO^{i,P}$ formula, with an *arbitrary order* i, to express the query,
2. • translate algorithmically the $HO^{i,P}$ formula into an SO formula,
3. • evaluate the SO formula.

Note that we have still (deterministic) single exponential time complexity, (NP complete queries *are still there*!) in the third step.

But *we don't have to deal* with *hyper exponential complexity.*

A Note on the Different Translations

The first translation (schema $\mathfrak{T}$ of TO) yields a *more clear and intuitive* SO formula, and the *maximum arity* of the quantified SO relation variables in general seems to be *much smaller*.

For the case of *hypercube graphs* the *maximum arity* obtained by the schema translation is 4, while for the SO formulas obtained by the $HO^{i,P}$ translation is 8.

And for the case of the *Formula-Value* query the *maximum arity* obtained by the schema translation is also 4, while for the SO formulas obtained by the $HO^{i,P}$ translation is 22.

Note that the *arity* of a relation symbol in an SO formula is relevant for the *complexity* of its evaluation (see among others [Hella, Turull-Torres,2006]).

Hence, and not surprisingly it *makes sense* to study *specific schemas* of TO formulas that have equivalent SO formulae, aiming to find *more efficient translations* than the general strategy used for $HO^{i,P}$ formulas (which *had the purpose of proving equivalence*, *rather than* looking for efficiency in the translation).

3: Valuating Relations of Poly-logarithmic Cardinality

A Query in TO^{plog}
Graph Factoring

[Ferrarotti, González, Schewe, Turull-Torres, 2018]

Roughly, let TO^{plog} denote the fragment of TO where only valuations which assign TO relations of *poly-logarithmic* cardinality, to TO variables are considered.

The SO sub-formulas in TO^{plog} are standard SO formulas.

For that matter we use typed TO variables of the form $\mathcal{X}^{\tau,\log^k}$, meaning that valuations can *only* assign to them relations of type τ and *cardinality* $\leq (\lceil \log n \rceil)^k$.

The input structure is $\mathbf{A}$ of signature $\sigma_F = \langle V_I, E_I, \mathcal{F}_I \rangle$, where $(V_I^{\mathbf{A}}, E_I^{\mathbf{A}})$ is a connected and loopless undirected graph (*cu-graph*), and $\mathcal{F}_I^{\mathbf{A}}$ is a TO relation which in turn consists of a set of pairs of graphs $(V_{\mathcal{F}_I}^{\mathbf{A}}, E_{\mathcal{F}_I}^{\mathbf{A}})$, and $(V_K^{\mathbf{A}}, E_K^{\mathbf{A}})$. The first graph of each pair is a *cu-graph*, and the second graph is a *clique*.

We define *graph factoring* as a decision problem. A σ_F-structure $\mathbf{A}$ is in the class GraphFactoring iff the third-order relation $\mathcal{F}_I^{\mathbf{A}}$ is *a factoring* of the graph $(V_I^{\mathbf{A}}, E_I^{\mathbf{A}})$, where the first graph of each pair in $\mathcal{F}_I^{\mathbf{A}}$ is a cu-graph that is a factor of the graph $(V_I^{\mathbf{A}}, E_I^{\mathbf{A}})$, and the size of the corresponding clique is the exponent.

A straightforward consequence of the definition of graph product is that the *size* of any *factoring circuit* $\mathcal{C}$ for a structure $\mathbf{A}$ is at most $2 \cdot \lceil \log(|V_I^{\mathbf{A}}|) \rceil$, and the size of the TO relation ${\mathcal{F}_I}^{\mathbf{A}}$ on any given $\mathbf{A} \in$ GraphFactoring is at most $\lceil \log(|V_I^{\mathbf{A}}|) \rceil$.

$$\varphi_{GF} \equiv \exists \mathcal{V}_{\mathcal{C}} \mathcal{E}_{\mathcal{C}} \Big($$

$$\text{“FactoringCircuitForG}_{\text{I}}(\mathcal{V}_{\mathcal{C}}, \mathcal{E}_{\mathcal{C}})$$

$$\wedge \text{NodesCUgraphs}(\mathcal{V}_{\mathcal{C}}, \mathcal{E}_{\mathcal{C}})$$

$$\wedge \text{RootsPrimeGraphs}_{\mathcal{C}} \wedge \text{RootsIn}\mathcal{F}_{\text{I}\mathcal{C}}$$

$$\wedge \text{SingleOutputG}_{\text{I}\mathcal{C}}\text{”} \Big)$$

where $(\mathcal{V}_{\mathcal{C}}, \mathcal{E}_{\mathcal{C}})$, is a TO graph of *size* at most $2 \cdot \lceil \log(|V_I^{\mathbf{A}}|) \rceil$, whose nodes are cu-graphs, and whose edges are pairs of cu-graphs.

$$\begin{aligned}
&\mathrm{FactoringCircuitForG_I}(\mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C}) \equiv \\
&\Big(\text{“Digraph}(\mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C}) \wedge \\
&\mathrm{Acyclic}(\mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C}) \wedge \\
&\mathrm{Connected}(\mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C}) \wedge \\
&\mathrm{InDegree2}_\mathcal{C} \wedge \\
&\mathrm{ProductOfParents}_\mathcal{C} \wedge \\
&\mathrm{LinearNonRoots}_\mathcal{C} \\
&\wedge \mathrm{NonIsomorphicRoots}_\mathcal{C}\text{”}\Big)
\end{aligned}$$

"InDegree$2_{\mathcal{C}}$" says that every node in the circuit has either 1 or 2 input nodes.

"ProductOfParents$_{\mathcal{C}}$" says that every node in $\mathcal{V}_{\mathcal{C}}$ is a *cu-graph* that is either the product of its *two parents*, or the square of its *single parent.*

$$\text{Product}(V_1, E_1, V_2, E_2, V_3, E_3) \equiv$$

$$\exists V_\times E_\times \Bigg(\Big[\forall v_1 w_1 v_2 w_2$$

$$\Big((V_\times(v_1, w_1) \leftrightarrow (V_1(v_1) \wedge V_2(w_1))) \wedge$$

$$\Big[E_\times(v_1, w_1, v_2, w_2) \leftrightarrow$$

$$\Big((v_1 = v_2 \wedge E_2(w_1, w_2)) \vee$$

$$(w_1 = w_2 \wedge E_1(v_1, v_2)) \Big) \Big] \Big) \Big] \wedge$$

$$\text{“Isomorphic}(V_\times, E_\times, V_3, E_3)\text{”} \Bigg)$$

$$\text{LinearNonRoots}_{\mathcal{C}} \equiv \exists \mathcal{V}_{\mathcal{C}l} \mathcal{E}_{\mathcal{C}l} \Big($$

$$\text{“EqualTO}\big(\mathcal{V}_{\mathcal{C}l}, \{\text{int. nodes in } \mathcal{C}\}\big) \wedge$$

$$\text{EqualTO}\big(\mathcal{E}_{\mathcal{C}l}, \mathcal{E}_{\mathcal{C}} \restriction \{\text{int. nodes in } \mathcal{C}\}\big)$$

$$\wedge \text{LinearDigraph}(\mathcal{V}_{\mathcal{C}l}, \mathcal{E}_{\mathcal{C}l})'' \Big)$$

where $\mathcal{E}_{\mathcal{C}} \restriction \{\text{int. nodes in } \mathcal{C}\}$ is the restriction of the TO binary relation $\mathcal{E}_{\mathcal{C}}$ to the subset of *internal nodes* of the set $\mathcal{V}_{\mathcal{C}}$.

$$\mathrm{NumbOfProducts}_{\mathcal{C}}(V_0, E_0, V_{K0}) \equiv$$

$$\exists \mathcal{H} \left(\text{“}\mathcal{H} : V_{K0} \mapsto \mathrm{Children}_{\mathcal{C}}(V_0, E_0) \; \textit{quasi} \text{ injective”} \right)$$

The *quasi injectivity* of the function in the formula above is due to the fact that we avoid allowing multiple edges between two given nodes in the circuit $\mathcal{C}$, to make the formula simpler.

Note that the *only* possible case where one *single edge* means that a (factor) graph is actually being used twice in the same product is at the (unique) node at *depth one* in the circuit.

An example for this situation is the factoring circuit for an *hypercube* of order n, where the same factor graph (K_2) is used n times.

Note:
As the *sizes* of the valuating TO relations that make the formula φ_{GF} true are *poly-logarithmic*, then it seems straightforward to apply the same encoding strategy as in $\mathrm{HO}^{i,P}$ and *translate* it to an SO formula.

Hence, we have the following:

Corollary:
$\mathrm{TO}^{plog} = \mathrm{SO}$.

Though the query graph factoring can certainly be *expressed* in SO (for instance with a signature

$$\sigma_F = \langle V_I^1, E_I^2, V_F^2, E_F^3, V_K^2, E_K^3 \rangle),$$

it doesn't seem to be easy.

Roughly, let SO^{plog} denote the fragment of SO where only valuations which assign SO relations of *poly-logarithmic* cardinality, to SO variables are considered.

For that matter we use typed SO variables of the form $X^{r,\log^k}$, meaning that valuations can only assign to them relations of arity r and cardinality $\leq (\lceil \log n \rceil)^k$.

And let $\mathrm{TO}^{plog}(\mathrm{SO}^{plog})$ denote the fragment of TO^{plog} where only valuations which assign SO relations of *poly-logarithmic* cardinality, to SO variables are considered.

Expected result:

With the same strategy, we *believe* that we can also prove:

- $\mathrm{TO}^{plog}(\mathrm{SO}^{plog}) = \mathrm{SO}^{plog}$.

[Ferrarotti, González, Schewe, Turull-Torres, 2018a]

On the other hand, we proved the following result:

- $\sum_1^{1,plog}(b\forall) = \text{NPolyLogTime}$.
- $\text{SO}^{plog} = \text{PLH}$. (*)

[(*) Barrington gave a characterization of the class of DCL-uniform families of Boolean circuits of unbounded fan-in, and quasi polynomial size (i.e., $2^{(\log n)^{O(1)}}$) and constant depth with an *equivalent* logic ([Barrington, 1992]). From that result the *second result* above follows.]

Where $\sum_1^{1,plog}(b\forall)$ is the existential fragment of SO^{plog} where the FO $\forall$ is bounded to *poly-logarithmic* sub-domains. And PLH denotes the non deterministic Polylog-Time Hierarchy.

Expected result:

Then, we *would* have also that:

- $\mathrm{TO}^{plog}(\mathrm{SO}^{plog}) = \mathrm{PLH}$.

This would mean that we can use a higher level language like TO^{plog} (SO^{plog}) to *prove* that a given query is in PLH.

That would make easier both the construction of the *formulas* and the corresponding *proofs*.

Examples in $\mathrm{TO}^{plog}(\mathrm{SO}^{plog})$**:**

- There is an induced subgraph $(V',$ $E')$ of size between $\lceil \log n \rceil$ and $(\lceil \log n \rceil)^c$, and there is a set $\mathcal{F}$ of size at least $(\lceil \log n \rceil)^{1/2}$, of disjoint induced subgraphs (V_i', E_i'), s. t. the subgraphs in $\mathcal{F}$ are a set of *prime factors* of the subgraph (V', E').
- There are between $\lceil \log n \rceil$ and $(\lceil \log n \rceil)^c$ disjoint induced subgraphs that are *cliques* of sizes between $\lceil \log n \rceil$ and $(\lceil \log n \rceil)^d$.

Note that the *first* query, doesn't seem to have an easy SO^{plog} formula.

To express it in $\mathrm{TO}^{plog}(\mathrm{SO}^{plog})$ we can follow a similar strategy as for Graph-Factoring above.

We *believe* that the following queries can be also expressed in $\mathrm{TO}^{plog}(\mathrm{SO}^{plog})$:

- *All* the induced *subgraphs* of size between $\lceil \log n \rceil$ and $(\lceil \log n \rceil)^c$ are *prime*.
- There are polylog disjoint induced *subgraphs* of polylog size s.t. for each of them, all its *prime factors* are *disjoint induced subgraphs* of size polylog.
- For every polylog size *set* of disjoint induced *subgraphs* of polylog size in G_1 there is a *set* of the same size of disjoint induced *subgraphs* of polylog size in G_2, s.t. there is a bijection $\mathcal{F} : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ so that the two graphs in ev-

ery pair in $\mathcal{F}$ are *isomorphic*.

So, proving that result, we would be able to use $\mathrm{TO}^{plog}(\mathrm{SO}^{plog})$ logic to write probably many queries in a *much simpler* way than using SO^{plog}.

And still, in that way *proving* that the queries are in PLH.

But we believe that we can do *better...*

Expected result:

Finally, we also *believe* that with the same strategy, we can prove:

- $\mathrm{HO}^{i,plog}(\mathrm{HO}^{<i,plog})$

  $= \mathrm{SO}^{plog} = \mathrm{PLH}.$

4: Beyond Second Order

SATQBF

SATQBF_k **and** SATQBF

QBF_k denotes the set of *quantified propositional formulas* of the form

$$\phi \equiv \exists \bar{x}_1 \forall \bar{x}_2 \dots Q\bar{x}_k(\varphi),$$

where φ is a propositional formula over $X = \{x_{ij}\}_{1 \leq i \leq k, 1 \leq j \leq l_i}$, $n \geq 0$, and where for $1 \leq i \leq k$, $\bar{x}_i = (x_{i1}, \dots, x_{il_i})$ is a tuple of l_i different variables from X.

Note that Q is "$\exists$" if k is odd and "$\forall$" if k is even, and the sets $X_1, \ldots, X_k$ of variables in $\bar{x}_1, \ldots, \bar{x}_k$, respectively, form a partition of X.

Let $\mathrm{QBF} = \bigcup_{k>0} \mathrm{QBF}_k$.

The semantics of the quantifiers is as follows: $\exists x(\alpha(x)) \equiv \alpha(0/x) \vee \alpha(1/x)$, and $\forall x(\alpha(x)) \equiv \alpha(0/x) \wedge \alpha(1/x)$.

Note that, in view of the semantics of the quantifiers, every quantified propositional formula is *equivalent to a propositional formula*, which is longer (roughly, *exponentially longer* in the number of quantifiers).

ϕ is *satisfiable* if *there is* a partial valuation $v_1 : X_1 \to \{T, F\}$, s. t. *for every* partial valuation $v_2 : X_2 \to \{T, F\}$, *there is* a partial valuation $v_3 : X_3 \to \{T, F\}$, s. t. ... s. t. the valuation $v = v_1 \cup v_2 \cup v_3 \cup \ldots \cup v_k$ makes φ true.

We now define *Boolean queries*:

- For $k > 0$, SATQBF_k is the set of quantified propositional formulas in QBF_k, represented as *word models* in the signature

  $$\langle \leq^2, I^1_x, I^1_\exists, I^1_\forall, I^1_\vee, I^1_\wedge, I^1_\neg,$$

  $$I^1_(, I^1_), I^1_| \rangle$$

  that are *true*.
- SATQBF is the set of quantified propositional formulas in QBF that are *true*.

Note that as formulas in QBF have no free variables, such a formula is *satisfiable* iff it is *true*.

SATQBF$_k$

- In Σ^1_k.

- It *doesn't look* like there is a simple *SO* formula to express SATQBF$_k$ *on word models* (see formula in [Ferrarotti, Ren, Turull-Torres, 2014]).

- [Pap,94] *Complete* for Σ^{P}_k under PTIME reductions.

SATQBF

- In Σ^2_1.
- It *doesn't look* like there is a simple TO formula to express SATQ BF *on word models* (see formula in [Ferrarotti, Ren, Turull-Torres, 2014]).
- [Imm,99,P.10.2] PSPACE *complete* via $(FO+ \leq +BIT)$ *reductions*.

SATQBF in HO^3 (known to be in HO^3)

At the *first* level of abstraction:

"There is a third order *alternating valuation* $\mathbf{T}_v$ *applicable* to φ, which satisfies φ".

At the *second* level of abstraction we express the following:

"$\exists$ *av* $\Delta^3 = (\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta)$";

"$\exists$ *linear* digraph $G_q = (V_q, E_q)$";

$\Big(\Big[$"$\mathcal{B}^3_\Delta : \mathcal{V}^3_\Delta \to \{0,1\}$"$\Big] \wedge$

$\Big[$"$G_q = (V_q, E_q)$ represents the *sequence* of quantified variables in φ"$\Big] \wedge$

$\Big[$"$(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta)$ is an *av applicable* to φ, i.e.":

- "$(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta)$ is a TO *binary tree* with all its *leaves* at the *same depth*, which is in turn equal to the *length* of (V_q, E_q)";

- "all the nodes in $(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta)$ whose *depth* correspond to a *universally* quantified variable in the prefix of quantifiers of φ, have exactly *one* sibling, and its value under $\mathcal{B}^3_\Delta$ is different than that of the given node";

- "all the nodes whose *depth* correspond to an *existentially* quantified variable in the prefix of quantifiers of φ, are either the *root* or have *no* siblings"$\Big]$

$\wedge$

$$\left[\text{“}(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta) \models_{av} \varphi\text{”} \right] \Big)$$

with $\models_{av}$ we denote that every *leaf* valuation of the *av* satisfies the quantifier-free sub formula of φ.

What about using HO^4?

Next we use an HO^4 formula instead.

We don't need to say that the *av* is *applicable* to φ; we just describe *how to build it*, which we believe is more *intuitive* and *simpler*.

Note that for the valuation of the fourth order variables it is enough if we consider *only* relations with cardinality $exp(1)$.

SATQBF **as a Sequence of** av**'s in** $\mathrm{HO}^{4,exp(1)} = \mathrm{HO}^3$ **(known to be in** HO^3**)**

At the *second* level of abstraction we express the following:

"$\exists$ sequence (of linear size) $(\mathcal{T}^4, \mathcal{E}^4)$ of av's $\Delta^3 = (\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta)$ (each av of size $exp(1)$)";

" $\exists$ bijection (of linear size) $\mathcal{F}^4_{\mathcal{T},\varphi}$: $\mathcal{T}^4 \rightarrow \{x : (I_\forall(x) \vee I_\exists(x))\}$ that preserves $\mathcal{E}^4$ and $\leq_\varphi$";

$$\Big[\text{“}\forall\, av\text{'s}\, \mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta,\ \mathcal{V}^3_{\Delta'}, \mathcal{E}^3_{\Delta'}, \mathcal{B}^3_{\Delta'}\text{”}$$

$$\Bigg(\Big[\text{“}\mathcal{B}^3_\Delta : \mathcal{V}^3_\Delta \to \{0,1\}\text{”}\Big]$$

$$\wedge$$

$$\Big[\text{“First}_{\mathcal{E}}(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta)\text{”} \to$$

$$\text{“}(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta) \text{ is an } av \text{ with just}$$

$$\textit{one} \text{ node”}\ \Big]$$

$$\wedge$$

$$\Big[\text{“Last}_{\mathcal{E}}(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta)\text{”} \to$$

$$\text{“}(\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta) \models_{av} \varphi\text{”}\ \Big]$$

$$\wedge$$

$$\Big[\text{“Succ}_{\mathcal{E}}(\mathcal{V}^3_{\Delta'}, \mathcal{E}^3_{\Delta'}, \mathcal{B}^3_{\Delta'}, \mathcal{V}^3_{\Delta}, \mathcal{E}^3_{\Delta}, \mathcal{B}^3_{\Delta})\text{”} \rightarrow$$

“*av* Δ is an *extension* of *av* Δ'

by *one* level in depth, so that”:

$\Big[$ "$I_{\exists}\big(\mathcal{F}^4_{\mathcal{T},\varphi}(\mathcal{V}^3_{\Delta}, \mathcal{E}^3_{\Delta}, \mathcal{B}^3_{\Delta})\big)$" $\rightarrow$
"each leaf of av $(\mathcal{V}^3_{\Delta'}, \mathcal{E}^3_{\Delta'}, \mathcal{B}^3_{\Delta'})$ has
exactly 1 child in its image in
$(\mathcal{V}^3_{\Delta}, \mathcal{E}^3_{\Delta}, \mathcal{B}^3_{\Delta})$, with an arbitrary
value in $\mathcal{B}^3_{\Delta}$" $\Big]$

$\wedge$

$\Big[$ "$I_{\forall}\big(\mathcal{F}^4_{\mathcal{T},\varphi}(\mathcal{V}^3_{\Delta}, \mathcal{E}^3_{\Delta}, \mathcal{B}^3_{\Delta})\big)$" $\rightarrow$
"each leaf of av $(\mathcal{V}^3_{\Delta'}, \mathcal{E}^3_{\Delta'}, \mathcal{B}^3_{\Delta'})$
has exactly 2 children in its image
in $(\mathcal{V}^3_{\Delta}, \mathcal{E}^3_{\Delta}, \mathcal{B}^3_{\Delta})$, with different
values in $\mathcal{B}^3_{\Delta}$" $\Big] \Bigg] \Bigg) \Bigg]$

where

"$av\ \Delta = (\mathcal{V}^3_\Delta, \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta)$ is an *extension* of $av\ \Delta' = (\mathcal{V}^3_{\Delta'}, \mathcal{E}^3_{\Delta'}, \mathcal{B}^3_{\Delta'})$"

is roughly expressed as follows:

"$\exists$ a total injection (of size $exp(1)$) $\mathcal{H}^3 : \mathcal{V}^3_{\Delta'} \to \mathcal{V}^3_\Delta$"

$$\left(\text{“}\mathcal{H}^3 \text{ preserves } \mathcal{E}^3_\Delta, \mathcal{B}^3_\Delta, \mathcal{E}^3_{\Delta'}, \mathcal{B}^3_{\Delta'}\text{”} \right)$$

Note that in the formula above for the valuation of the 4-th *order* variables it is *enough* if we consider *only* relations of cardinality $exp(1)$.

We could then *encode* the HO^4 relations in TO relations, using tuples of (SO) *sets* as identifiers for *tuples* of TO relations in the 4-th *order* relations.

Expected result:

For every $i \geq k \geq j \geq 4$ let $\mathrm{HO}^{i,exp(j-2)}$ denote the fragments of HO^{i} where the *cardinality* of the valuating k-th order relations for the k-th *order* variables are restricted to be $O(exp(j-2))$ w.r.t. the size of the model.

Then, we *believe* that, by using basically the same *encoding strategy* as for $\mathrm{HO}^{i,P}$, we can prove the following:

- For every $i \geq k \geq j \geq 4$:

  $\mathrm{HO}^{i,exp(j-2)}$ *collapses* to HO^{j}.

In the encoding of relations of order k as above, the difference w.r.t. $\mathrm{HO}^{i,P}$ is that we need *more different identifiers* to *encode* tuples of HO^{k-1} relations.

Note that, as the cardinality of an HO^{k} relation is at most $exp(k-2)$, the number of *different* HO^{k} *relations* is at most $exp(k-1)$.

Then, to *encode* HO relations, of whichever order k, whose *maximum cardinality* is $O(exp(j-2))$,

we need $O(exp(j-2))$ different *identifiers*, and hence a tuple of relations of order $(j-1)$ is enough.

So that in the *db that encodes* a relation of *order* k:

- all the relations will use *tuples of relations* of order $(j-1)$ as *identifiers* for tuples of relations of order $k-1$,

- and hence, relations of *order* j *suffice* to represent the db.

5: Beyond Third Order

$$\mathrm{SATQBF}(\Sigma_k^i)$$

We will next see an example of a query known to be expressible in HO^4.

It *doesn't seem easy* to express it in HO^4.

We will use HO^6 and HO^7 to express it instead.

And then we will see that we (*believe* that) we can translate *both* formulas to HO^5.

A more complex problem: High Order SATQBF_k [Hella, Turull-Torres, 2006]

We want to build a variant of the problem SATQBF_k of a higher complexity, that is, a *higher order* variant, considering the logics Σ^i_j for all $i, k \geq 1$.

But we must remain as close as possible to *propositional logic*.

With that in mind, we consider *one single structure*, that we call the *Boolean model*,

$$\mathcal{B} = \langle \{a, b\}, 0^{\mathcal{B}}, 1^{\mathcal{B}} \rangle$$

a *two-element* model where both elements are interpretations of the constant symbols 0 and 1.

Then, deciding whether a given Σ^i_j sentence is "*satisfiable*" (in the *propositional logical* sense), turns into deciding whether a Σ^i_j sentence in the vocabulary of the Boolean model, is *true* in the Boolean model.

That is, it means deciding the Σ^i_j *theory* of the Boolean model:

$$\Sigma^i_j\text{-Th}(\mathcal{B}).$$

The problem $\mathrm{SATQBF}(\Sigma_k^i)$

For $i, k \geq 1$ let $\mathrm{SATQBF}(\Sigma_k^i)$ denote the Boolean query:

"given a Σ_k^i sentence ϕ in the vocabulary of the *Boolean model*, is $\phi \in \Sigma_k^i\text{-Th}(\mathcal{B})$?".

***Descriptive* Complexity of** $\mathrm{SATQBF}(\Sigma^i_k)$ [Hella, Turull-Torres, 2006]

Then we have the following:

- For $i, k \geq 1$, $\mathrm{SATQBF}(\Sigma^i_k)$ *on word models* is *complete* for Σ^{i+1}_k under P reductions.

Note that each Σ^i_k sentence is represented as a *string* in the alphabet of predicate logic of order i.

Note that the notion of *completeness* of the result above is *w.r.t. a logic*, not to a (computational) complexity class, i.e., it is a notion in the setting of *descriptive complexity*.

This means that for every Σ_k^{i+1} sentence ψ of an *arbitrary* vocabulary τ, and *every* τ-structure $\mathcal{A}$, we build (in polynomial time) a Σ_k^i sentence $f_\psi(\mathcal{A})$ on the Boolean model, s. t.

$$\mathcal{B} \models f_\psi(\mathcal{A}) \text{ iff } \mathcal{A} \models \psi.$$

Computational **Complexity of** $\mathrm{SATQBF}(\Sigma^i_k)$ [Hella, Turull-Torres, 2006]

Considering the expressibility of Σ^{i+1}_k given above, we also get:

- For $i \geq 1$ and $k \geq 1$, $\mathrm{SATQBF}(\Sigma^i_k)$ *on word models* is *complete* for NEXPH^{k-1}_i under P reductions.

Note that these problems being *complete* for NEXPH_i^{k-1}, implies that they are *provably intractable*, that is, we *know* that for each $i \geq 1$ and $k \geq 1$, *there is no* algorithm in P that can decide $\mathrm{SATQBF}(\Sigma_k^i)$.

This is because there are *provably intractable problems* in $\mathrm{NTIME}(2^{n^c})$, and hence all the classes that include it contain intractable problems too [Garey,Johnson,1979].

The problems $\mathrm{SATQBF}(\Sigma_k^i)$ are the *first* known family of complete problems for all the levels of the Non deterministic Hyper-exponential Time Hierarchy NEXPH_i^{k-1}.

SATQBF(Σ^2_j)

In the word model for the input formula $\varphi \in \Sigma^2_j$, the variables and their *types* are encoded as follows (where $Q \in \{\exists, \forall\}$, and $i, r_i, t_i \geq 1$):

- 1st order variable x_i:

  $Qx|^i$

- 2nd order variable R_i of arity r_i:

  $QR|^i * |^{r_i}$

- 3nd order variable $\mathcal{R}_i$ of type $\tau_i = (r_1, \ldots, r_{t_i})$:

  $Q\mathcal{R}|^i * (|^{r_1}, \ldots, |^{r_{t_i}})$

The signature of the word model is the following:

$$\langle \leq^2, I^1_{\mathcal{R}}, I^1_R, I^1_x, I^1_{\exists}, I^1_{\forall}, I^1_{\vee}, I^1_{\wedge}, I^1_{\neg}, I^1_{(}, I^1_{)}, I^1_{,}, I^1_{|}, I^1_{*} \rangle$$

We assume that the *quantifier blocks* are arranged in the order ⟨3rd, 2nd, 1st⟩ order quantifiers, and are then followed by a *quantifier free* formula.

The *first* quantifier is always a 3rd order *existential* quantifier.

Representation of HO Relations SO variables

An r-ary SO variable $S^{2,r}$:

as a TO relation $\mathcal{S}^{3,\tau^2}$, with $\tau^2 = (1,2,2)$, i.e., a *set of linear digraphs* of size r with a Boolean assignment.

So that *each* such digraph represents an *r-tuple* in the SO relation that valuates $S^{2,r}$.

Representation of HO Relations TO variables

A TO variable $\mathcal{R}^{3,\tau^3}$ of type $\tau^3 = (r_1, \dots, r_s)$:

as an HO^5 relation $\mathcal{R}^{5,\tau^5}$.

In the TO relation that valuates $\mathcal{R}^{3,\tau^3}$:

- *each tuple* of SO relations has s components which are SO relations of arities $r_1, \ldots, r_s$, respectively;
- hence, *each such tuple* is represented in $\mathcal{R}^{5,\tau^5}$ as a sequence of linear digraphs with Boolean assignments,

- that is, it is an HO^4 *linear digraph* of size s where *each node* is a *TO set* of linear (SO) digraphs of sizes $r_1, \ldots, r_s$, respectively;

then, a TO relation, i.e., a set of *tuples* of *SO relations*, is represented in $\mathcal{R}^{5,\tau^5}$ as a set of HO^4 linear digraphs of size s, hence as an HO^5 relation.

$$\tau^5 =$$

$$\Big(\big((1,2,2)\big), \big((1,2,2),(1,2,2)\big)\Big)$$

SATQBF(Σ^2_j)
in $\mathrm{HO}^{6,exp(3)} = \mathrm{HO}^5$
(known to be in HO^4**)**

"$\exists\, av\, \Delta^6 = (\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta)$ (of size $exp(3)$)";

"$\exists$ *linear* digraph $G_q = (V_q, E_q)$"

that represents de sequence of *quantified* variables in φ, ordered as $\langle$3rd, 2nd, 1st$\rangle$ order variables;

"$\exists\; F_{q,\varphi}$:

$V_q \rightarrow \{z : (I_x(z) \vee I_R(z) \vee I_{\mathcal{R}}(z))\}$

total bijection (of linear size) that preserves E_q and $\leq_\varphi$";

“$\exists\, \mathcal{F}^6_{\Delta,q} : \mathcal{V}^6_\Delta \rightarrow V_q$ total surjective function (of size $exp(3)$) that maps every node in $av\ \Delta^6$ to its corresponding quantified variable in φ”;

$$\Bigg(_1 \Big[\text{“}(\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta) \text{ is an out-tree with}$$

$$\text{all leaves at depth } |V_q|\text{”}\Big] \wedge$$

$$\Big[\text{“}\mathcal{V}^6_\Delta \text{ is a set of tuples } (\mathcal{I}^5, x^1, \mathcal{S}^3, \mathcal{R}^5)\text{”}\Big]$$

$$\wedge\ \forall z \Bigg[_2 V_q(z) \rightarrow \Bigg(_3 \Bigg[_4$$

$$\Bigg(_5 \text{“}I_{\mathcal{R}}\big(F_{q,\varphi}(z)\big) \wedge$$

$$I_\exists\big(Pred_{\leq\varphi}(F_{q,\varphi}(z))\big)\text{”} \Bigg)_5 \rightarrow$$

$$\Bigg(_5 \Big[_6 \text{“First}_{E_q}(z) \wedge \exists\, \mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5\text{”}$$

$$\Big(\text{“Root}_\Delta(\mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5) \wedge$$

$$\mathcal{S}_1^3 = \emptyset \ \wedge x_1^1 = 3 \wedge$$

'$\mathcal{R}_1^5$ is well formed as a

representation of a 3rd order

relation' ” $\Big)\Big]_6$

$\vee$

$$\Big[_6 \neg \mathrm{First}_{E_q}(z) \wedge$$

$$\forall\, \mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5, \quad \exists\, \mathcal{I}_2^5, x_2^1, \mathcal{S}_2^3, \mathcal{R}_2^5$$

$$\Big(_7 \Big[\text{“}\mathcal{F}_{\Delta,q}(\mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5) = \mathrm{Pred}_{E_q}(z)\text{”}\Big]$$

$$\rightarrow$$

$$\Big[\text{“}(\mathcal{I}_2^5, x_2^1, \mathcal{S}_2^3, \mathcal{R}_2^5) \text{ is the } \textit{unique} \text{ child}$$

$$\text{of } (\mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5) \text{ in } av\ \Delta^6\text{”} \ \wedge$$

$$\mathcal{S}_2^3 = \emptyset \ \wedge \ x_2^1 = 3 \ \wedge$$

$$\text{'}\mathcal{R}_2^5 \text{ is well formed...' ”}\Big]\Big)_7\Big]_6\Bigg)_5\Bigg]_4$$

$$\wedge$$

$$\Big[_4 \text{``}I_{\mathcal{R}}\big(F_{q,\varphi}(z)\big)\wedge I_{\forall}\big(Pred_{\leq_\varphi}(F_{q,\varphi}(z))\big)\text{''}$$

$$\rightarrow \ldots\Big]_4$$

$$\wedge$$

$$\Big[_4 \text{``}I_{R}\big(F_{q,\varphi}(z)\big)\wedge I_{\exists}\big(Pred_{\leq_\varphi}(F_{q,\varphi}(z))\big)\text{''}$$

$$\rightarrow \ldots\Big]_4$$

$$\wedge$$

$$\Big[_4 \text{``}I_{R}\big(F_{q,\varphi}(z)\big)\wedge I_{\forall}\big(Pred_{\leq_\varphi}(F_{q,\varphi}(z))\big)\text{''}$$

$$\rightarrow \ldots\Big]_4$$

$$\wedge$$

$$\Big[_4 \text{``}I_{x}\big(F_{q,\varphi}(z)\big)\wedge I_{\exists}\big(Pred_{\leq_\varphi}(F_{q,\varphi}(z))\big)\text{''}$$

$$\rightarrow \ldots\Big]_4$$

$$\wedge$$

$$\Big[_4 \text{``}I_x\big(F_{q,\varphi}(z)\big)\wedge I_\forall\big(Pred_{\leq\varphi}(F_{q,\varphi}(z))\big)\text{''}$$

$$\rightarrow \ldots\Big]_4$$

$$\Bigg)_3\Bigg]_2$$

$$\wedge$$

$$\forall\, \mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5$$

$$\Big[_2 \text{``Leaf}_\Delta(\mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5) \rightarrow$$

$\Big(_3$ "the valuation in the path from the root of $av\ \Delta$ to the leaf $(\mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5)$ satisfies the q-free sub-formula of φ" $\Big)_3\Big]_2\Bigg)_1$

Note that with *each leaf valuation* we can build a *propositional* formula in $\{F, T\}$ from the q-free sub-formula of φ.

Each *TO atomic formula* in φ $\mathcal{R}^{3,\tau}(S_1, \ldots, S_{|\tau|})$ is replaced with the *truth value* of the fact that the tuple of *SO* relations assigned to the *SO* variables $S_1, \ldots, S_{|\tau|}$, belongs to the *TO* relation assigned to $\mathcal{R}^{3,\tau}$.

And we proceed similarly for the *SO atomic formulas*.

Then, to *evaluate* the resulting formula, we can use the *TO* formula in the fragment $\mathfrak{T}$ for the Formula-Value query mentioned above.

SATQBF(Σ^2_j) as a Sequence of av's in $\mathrm{HO}^{7,exp(3)} = \mathrm{HO}^5$ (known to be in HO^4)

"$\exists$ sequence (of linear size) $(\mathcal{V}^7_S, \mathcal{E}^7_S)$ of av's $\Delta^6 = (\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta)$ out-trees of size $exp(3)$, and depth growing from 1 to $|V_q|$";

"$\exists$ *linear* digraph $G_q = (V_q, E_q)$" that represents de sequence of *quantified* variables in φ, ordered as $\langle$3rd, 2nd, 1st$\rangle$ order variables;

"$\exists$ bijection (of linear size) $\mathcal{F}^7_{\mathcal{V}_S,\varphi}$: $\mathcal{V}^7_S \rightarrow \{x : (I_x(z) \vee I_R(z) \vee I_{\mathcal{R}}(z))\}$ that preserves $\mathcal{E}^7_S$ and $\leq_\varphi$, and maps every *av* Δ^6 to its corresponding quantifier in φ";

$\Big(_1 \Big[$"$\mathcal{V}^6_\Delta$ is a set of tuples $(\mathcal{I}^5, x^1, \mathcal{S}^3, \mathcal{R}^5)$"$\Big]$

$\wedge$ "$\forall$ *av*'s $\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta, \mathcal{V}^6_{\Delta'}, \mathcal{E}^6_{\Delta'}$" $\Big(_2$

$$\Big[_3 \text{“First}_{\mathcal{E}^7_S}(\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta)\text{”} \rightarrow$$

$$\Big(_4 \text{“}(\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta) \text{ is an } av \text{ with just}$$

$$one \text{ node } (\mathcal{I}^5, x^1, \mathcal{S}^3, \mathcal{R}^5)\text{”}$$

$$\wedge$$

$$\text{“}\mathcal{S}^3 = \emptyset \ \wedge x^1 = 3 \wedge$$

$$\text{’}\mathcal{R}^5_1 \text{ is well formed as a}$$

$$representation \text{ of a 3rd order}$$

$$\text{relation’ ”}\Big]\Big)_4\Big]_3 \wedge$$

$$\Big[_3 \text{“}Succ_{\mathcal{E}^7_S}(\mathcal{V}^6_{\Delta'}, \mathcal{E}^6_{\Delta'}, \mathcal{V}^6_{\Delta}, \mathcal{E}^6_{\Delta})\text{”} \rightarrow$$

“*av* Δ is an *extension* of *av* Δ'

by *one* level in depth, so that”:

$$\Big(_4 \Big[_5 \Big(_6 \text{“}I_\exists\big(\mathrm{Pred}_{\leq\varphi}(\mathcal{F}^7_{\mathcal{V}_S,\varphi}(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta))\big)\wedge$$

$$I_\mathcal{R}\big(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta\big)\text{”}\ \Big)_6 \rightarrow$$

$$\forall\, \mathcal{I}^5_1, x^1_1, \mathcal{S}^3_1, \mathcal{R}^5_1, \quad \exists\, \mathcal{I}^5_2, x^1_2, \mathcal{S}^3_2, \mathcal{R}^5_2$$

$$\Big(_6 \text{ “Leaf}_{\Delta'}(\mathcal{I}^5_1, x^1_1, \mathcal{S}^3_1, \mathcal{R}^5_1)\text{”} \rightarrow$$

$$\Big[\text{“}(\mathcal{I}^5_2, x^1_2, \mathcal{S}^3_2, \mathcal{R}^5_2) \text{ is the } \textit{unique} \text{ child}$$

$$\text{of the image of } (\mathcal{I}^5_1, x^1_1, \mathcal{S}^3_1, \mathcal{R}^5_1)$$

$$\text{in } av\ \Delta^6\text{”} \wedge \text{“}\mathcal{S}^3_2 = \emptyset \wedge x^1_2 = 3 \wedge$$

$$\text{'}\mathcal{R}^5_2 \text{ is well formed...' ”}\Big]\Big)_6\Big]_5 \wedge$$

$$\Big[_5 \text{``}I_\forall\Big(\mathrm{Pred}_{\leq\varphi}(\mathcal{F}^7_{\mathcal{V}_S,\varphi}(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta))\wedge$$

$$I_\mathcal{R}\big(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta\big)\text{''}\ \Big)_6 \to \ldots\Big]_5 \wedge$$

$$\Big[_5 \text{``}I_\exists\Big(\mathrm{Pred}_{\leq\varphi}(\mathcal{F}^7_{\mathcal{V}_S,\varphi}(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta))\wedge$$

$$I_R\big(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta\big)\text{''}\ \Big)_6 \to \ldots\Big]_5 \wedge$$

$$\Big[_5 \text{``}I_\forall\Big(\mathrm{Pred}_{\leq\varphi}(\mathcal{F}^7_{\mathcal{V}_S,\varphi}(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta))\wedge$$

$$I_R\big(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta\big)\text{''}\ \Big)_6 \to \ldots\Big]_5 \wedge$$

$$\Big[_5 \text{``}I_\exists\Big(\mathrm{Pred}_{\leq\varphi}(\mathcal{F}^7_{\mathcal{V}_S,\varphi}(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta))\wedge$$

$$I_x\big(\mathcal{V}^6_\Delta,\mathcal{E}^6_\Delta\big)\text{''}\ \Big)_6 \to \ldots\Big]_5 \wedge$$

$$\Big[_5 \text{“}I_\forall\Big(\text{Pred}_{\leq\varphi}(\mathcal{F}^7_{\mathcal{V}_S,\varphi}(\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta))\wedge$$

$$I_x\big(\mathcal{V}^6_\Delta, \mathcal{E}^6_\Delta\big)\text{”}\ \Big)_6 \to \dots\Big]_5\ \Bigg)_4\ \Bigg]_3$$

$$\wedge$$

$$\Big[_3 \text{“Last}_{\mathcal{E}_S^7}(\mathcal{V}_\Delta^6, \mathcal{E}_\Delta^6)\text{”} \rightarrow$$

$$\forall\, \mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5$$

$$\Big(_4 \text{“Leaf}_\Delta(\mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5) \rightarrow$$

$\Big[_5$ “the valuation in the path from the root of $av\ \Delta$ to the leaf $(\mathcal{I}_1^5, x_1^1, \mathcal{S}_1^3, \mathcal{R}_1^5)$ satisfies the q-free sub-formula of φ” $\Big]_5 \Big)_4 \Big]_3 \Big)_2 \Big)_1$

References

[Boerger, 2003] E. Börger, R. F. Stärk, "Abstract State Machines. A Method for High-Level System Design and Analysis", Springer, 2003.

[Bollobás, 2002] B. Bollobás, "Modern Graph Theory", Springer, Graduate Texts in Mathematics, 184, 2002.

[Downey, Fellows, 1999] R. G. Downey, M. R. Fellows, "Parameterized Complexity", Springer, Monographs in Computer Science, 1999.

[Ferrarotti, González, Turull-Torres,2017] F. Ferrarotti, S. González, J. M. Turull Torres, "On Fragments of Higher Order Logics that on Finite Structures Collapse to Second Order", Logic, Language, Information, and Computation, 24th International Workshop (WoLLIC 2017),Lecture Notes in Computer Science, 10388, Springer, J. Kennedy and Ruy de Queiroz, 125-139, 2017.

[Ferrarotti, González, Schewe, Turull-Torres,2018] F. Ferrarotti, S. González, K.-D. Schewe, J. M. Turull-Torres, "Systematic Refinement of Abstract State Machines with Higher-Order Logic", 6th International ABZ Conference ASM, Alloy, B, TLA, VDM, Z, June 5th-8th, 2018, Southampton, UK.

[Ferrarotti, González, Schewe, Turull-Torres, 2018a]

F. Ferrarotti, S. González, K.-D. Schewe, J. M. Turull-Torres, “The Polylog-Time Hierarchy Captured by Restricted Second-Order Logic”, CoRR, vol. abs/1806.07127, 2018. [Online]. Available: https://arxiv.org/abs/1806.07127

[Ferrarotti, Ren, Turull-Torres, 2014] F. Ferrarotti, W. Ren, J. M. Turull-Torres, “Expressing properties in second- and third-order logic: hypercube graphs and SATQBF”, Logic Journal of the IGPL, 22, 355-386, 2, 2014.

[Garey,Johnson,1979] Garey, M. R., Johnson, D. S. 1979. “Computers and Intractability - A guide to the Theory of NP-Completeness”. W. H. Freeman and Co., San Francisco, Calif.

[Hella,Turull-Torres,2006] Hella, L., Turull Torres, J. M., “Computing queries with higher order logics”, TCS 355, 2006.

[Hella,Turull-Torres,2006a] Hella, L., Turull Torres, J. M., “Complete Problems for Higher Order Logics”, in “Computer Science Logic 2006, Proceedings”, Springer, LNCS 4207, pp. 380-394, 2006.

[Immerman, 1999] N. Immerman, “Descriptive Complexity”, Springer, Graduate texts in computer science, 1999.

[Schellhorn,Ernst,Pfhler,Bodenmller, Reif , 2018] G.

Schellhorn, G. Ernst, J. Pfhler, S. Bodenmller, W. Reif, “Symbolic execution for a clash-free subset of ASMs”, in “Abstract State Machines, Alloy, B, TLA, VDM and Z (ABZ 2016)”, Ed. by M. Butler, K.-D. Schewe, Science of Computer Programming, 158, 21-40, https://doi.org/10.1016/j.scico.2017.08.014.